

# Task 7: Research Activity

## Part 1: Constructors

A constructor is a special function inside a class. It runs every time we create a new object. Constructors help us set starting values so the object works right away. This makes our code safer because the object will not start empty or broken. It also makes our program easier to fix later, because if we want to change how an object starts, we only change the constructor.

## Real-World Examples

- Games: When a new player or enemy is created, the constructor gives them default health, speed, and level.
- Bank Applications: When a new bank account object is made, the constructor sets the account number and starting balance.
- Web or API Applications: When we connect to a website or API, the constructor sets the base URL and key so we don't write it again.

## Sources Used

- Microsoft C# Documentation  
<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/constructors>
- W3Schools  
[https://www.w3schools.com/cs/cs\\_constructors.php](https://www.w3schools.com/cs/cs_constructors.php)
- YouTube video about C# Object-Oriented Programming  
<https://www.youtube.com/watch?v=GhQdIIFYlQ8>

## Part 2: OOP Principle – Encapsulation

Encapsulation is an object-oriented programming principle that focuses on protecting data within a class. By making certain variables private direct modifications from outside the class are prevented. Methods such as SetName() or GetBalance() are provided to allow controlled access and modification of the data. This approach helps keep programs secure and maintainable as it restricts how data is manipulated and ensures that only valid changes are made.

## **Classes and Objects**

A class serves as a blueprint or design for creating objects. When you create an object from a class, it becomes a real instance that can be used in your program.

- Class: The design or plan (e.g., a Dog blueprint).
- Object: The actual item created from that plan (e.g., a real dog).

## **Simple Examples**

- Bank System: The balance is kept private within the class. Users can only add or withdraw money through specific methods ensuring the balance is changed correctly.
- Student System: Student marks are stored as private variables. They can only be updated using methods preventing invalid values from being added.